

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

HUN SIK KANG, ET AL.

Application No.:

Filed:

For: **constellation mapping apparatus and method**

Art Group:

Examiner:

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR PRIORITY

Sir:

Applicant respectfully requests a convention priority for the above-captioned application, namely:

COUNTRY	APPLICATION NUMBER	DATE OF FILING
Korea	2002-0061788	10 October 2002

☒ A certified copy of the document is being submitted herewith.

Respectfully submitted,

Blakely, Sokoloff, Taylor & Zafman LLP

Dated: 10/8/03

12400 Wilshire Blvd., 7th Floor
Los Angeles, California 90025
Telephone: (310) 207-3800


Eric S. Hyman, Reg. No. 30,139

KOREAN INTELLECTUAL PROPERTY OFFICE

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

Application Number:: Korean Patent Application 2002-0061788

Date of Application:: 10 October 2002

Applicant(s) : Electronics and Telecommunications Research Institute

04 November 2002

COMMISSIONER

[Bibliography]

[Document Name]	Patent Application
[Classification]	Patent
[Receiver]	Commissioner
[Reference No.]	0003
[Filing Date]	10 October 2002
[IPC]	H04N
[Title]	Apparatus and method for constellation mapping
[Applicant]	
[Name]	Electronics and Telecommunications Research Institute
[Applicant code]	3-1998-007763-8
[Attorney]	
[Name]	Youngpil Lee
[Attorney code]	9-1998-000334-6
[General Power of Attorney Registration No.]	2001-038378-6
[Attorney]	
[Name]	Haeyoung Lee
[Attorney code]	9-1999-000227-4
[General Power of Attorney Registration No.]	2001-038396-8
[Inventor]	
[Name]	KANG, Hun Sik
[Resident Registration No.]	690817-1683710
[Zip Code]	305-345
[Address]	16/5, 263-1 Shinsung-dong, Yusong-gu, Daejeon-city Rep. Of Korea
[Nationality]	Republic of Korea
[Inventor]	
[Name]	KIM, Jong Won
[Resident Registration No.]	570115-1002424
[Zip Code]	302-120
[Address]	109-703 Hyangchon Apt. 970 Dunsan-dong, Seo-gu Daejeon-city, Rep. Of Korea
[Nationality]	Republic of Korea
[Request for Examination]	Requested

[Purpose] We file as above according to Art. 42 of the Patent Law, request the examination as above according to Art. 60 of the Patent Law.

Attorney

Young-pil Lee

Attorney

Hae-young Lee

[Fee]

[Basic page] 20 Sheet(s) 29,000 won

[Additional page] 21 Sheet(S) 21,000 won

[Priority claiming fee] 0 Case(S) 0 won

[Examination fee] 15 Claim(s) 589,000 won

[Total] 639,000 won

[Reason for Reduction] Government Invented Research Institution

[Fee after Reduction] 319,500 won

[Transfer of Technology] Allowable

[Licensing] Allowable

[Technology Training] Allowable

[Enclosures]

1. Abstract and Specification (and Drawings) 1 copy

대한민국 특허청

KOREAN INTELLECTUAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2002-0061788
Application Number PATENT-2002-0061788

출원년월일 : 2002년 10월 10일
Date of Application OCT 10, 2002

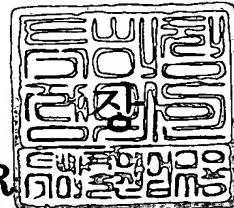
출원인 : 한국전자통신연구원
Applicant(s) Electronics and Telecommunications Research Institute



2002 년 11 월 04 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0003
【제출일자】	2002.10.10
【국제특허분류】	H04N
【발명의 명칭】	성좌 매핑 장치 및 매핑방법
【발명의 영문명칭】	Apparatus and method for constellation mapping
【출원인】	
【명칭】	한국전자통신연구원
【출원인코드】	3-1998-007763-8
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2001-038378-6
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2001-038396-8
【발명자】	
【성명의 국문표기】	강현식
【성명의 영문표기】	KANG,Hun Sik
【주민등록번호】	690817-1683710
【우편번호】	305-345
【주소】	대전광역시 유성구 신성동 263-1 16/5
【국적】	KR
【발명자】	
【성명의 국문표기】	김종원
【성명의 영문표기】	KIM,Jong Won
【주민등록번호】	570115-1002424

【우편번호】	302-120		
【주소】	대전광역시 서구 둔산동 970번지 향촌아파트 109-703		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 이영필 (인) 대리인 이해영 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	21	면	21,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	15	항	589,000 원
【합계】	639,000 원		
【감면사유】	정부출연연구기관		
【감면후 수수료】	319,500 원		
【기술이전】			
【기술양도】	희망		
【실시권 허여】	희망		
【기술지도】	희망		
【첨부서류】	1. 요약서·명세서(도면)_1통		

【요약서】**【요약】**

본 발명은 작은 크기의 메모리와 간단한 디코딩 회로를 사용하여 종래의 성좌 처리기의 복잡도를 줄이고, 여러개의 성좌를 처리할 수 있는 성좌 처리 장치에 관한 것이다. 본 발명의 성좌 처리 장치는 성좌 점 데이터, 비트 당 심벌(Bits-per-Symbol) 정보, 심벌 유효 정보를 입력받아 상기 성좌 점에 대한 성좌값을 저장하고 있는 메모리의 주소 정보를 발생시키는 주소 생성부와, 상기 발생된 주소 정보에 대응되며, 성좌의 사분면중 어느 사분면에 있는 성좌점을 저장하고 있는 메모리와, 상기 메모리에서 출력된 데이터를 반전 또는 비반전시키는 반전 회로부 및 스케일러부로 구성되어, 모든 성좌 점들을 메모리에 저장하지 않고 각 성좌도의 어느 한 사분면 내의 성좌 점들만 저장함으로써, 메모리의 크기를 줄일 수 있는 효과가 있다.

【대표도】

도 6

【명세서】**【발명의 명칭】**

성좌 매핑 장치 및 매핑방법{Apparatus and method for constellation mapping}

【도면의 간단한 설명】

도 1은 종래의 성좌 매핑 장치의 블록도이다.

도 2는 4-QAM과 16-QAM의 성좌 매핑도를 나타낸 도면이다.

도 3은 성좌 매핑을 위한 메모리 구성을 나타낸 도면이다.

도 4A 내지 도 4G는 본 발명에서 처리하는 성좌의 종류를 나타낸 도면이다.

도 5는 본 발명의 성좌 매핑 장치의 간략한 구성을 나타낸 블록도이다.

도 6은 본 발명의 성좌 매핑 장치의 상세한 구성을 나타낸 도면이다.

도 7은 비트 당 심벌값에 따른 유효 입력 데이터를 나타낸 도표이다.

도 8은 본 발명의 성좌 매핑 장치에서 사용되는 메모리의 구성도를 나타낸 도면이다.

도 9는 입력받은 데이터를 이동시킨 후의 값을 나타낸 도면이다.

도 10A 내지 도 10D는 반전 및 비반전의 선택신호의 발생에 관한 진리표와 논리식을 나타낸 도면이다.

도 11은 반전 제어부를 나타낸 도면이다.

도 12는 본 발명의 성좌 매핑 방법을 설명한 플로우 차트이다.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<13> 본 발명은 작은 크기의 메모리와 간단한 디코딩 회로를 사용하여 종래의 성좌 처리기의 복잡도를 줄이고, 여러개의 성좌를 처리할 수 있는 성좌 처리장치에 관한 것이다. 구체적으로는 데이터 통신에서 주로 사용되는 직교 변조기 구성의 하나로 쓰이는 성좌 매핑 장치에 관한 것으로 맥내 전화선을 이용한 고속 데이터 통신을 하고자 하는 분야에 이용될 수 있다.

<14> 인터넷의 보급에 따라 고속 데이터 통신이 증가하고 있는 가운데 맥내의 전화선을 이용하여 데이터 전송 속도를 고속화하려는 연구가 많이 진행되고 있다. 고속의 전송 속도를 실현하기 위해서 직교변조(Quadrature Amplitude Modulation, QAM)방식이 많이 사용되고 있다. 그리고, 직교변조(QAM) 방식을 사용하는 비대칭 디지털 가입자 회선(Asymmetric Digital Subscriber Line, ADSL)이나 홈 PNA(Home PhoneLine Network Alliance)와 같은 방식의 통신에서는 하나의 고정된 전송 속도로 통신을 하지 않고, 전화선의 채널 상황에 따라 전송속도를 다르게 하는 방식을 사용하고 있다.

<15> 채널상황에 따라 전송속도를 다르게 하기 위해서는 여러 개의 성좌 맵(Constellation Map)을 처리할 수 있는 직교변조기가 필요하다. 직교변조기는, 채널상황이 양호할 때에는 전화 선로를 통해 전송하는 하나의 심벌(symbol)에 많은 양의 비트정보를 대응시켜 전송하고, 그렇지 않을 경우에는 적은 양의 비트정보를 대응시켜 보낸다.

따라서, 직교변조기는 채널 상황에 따라서 낮은 전송속도를 갖는 성좌 매핑을 통해 정보를 보내거나 높은 전송속도를 갖는 성좌 매핑을 통해 정보를 보내야 한다.

<16> 참고로, 홈 PNA와 같은 방식의 통신에서는 한 심벌(symbol)당 2비트(bit)에서 8비트까지의 정보를 보낼 수 있는 방법을 사용하고 있는데, 이때 QPSK(Quadrature Phase Shift Keying), 8-QAM, 16-QAM, 32-QAM, 64-QAM, 128-QAM 및 256-QAM에 이르는 7개의 성좌를 통해 비트 정보를 보내도록 하고 있다. 이렇게 다중의 성좌를 이용하기 때문에 다른 크기의 성좌를 비트 매핑(bit-mapping)할 경우에 많은 양의 메모리를 필요로 한다. 또한, 전송하고자 하는 속도에 맞는 성좌로 입력 데이터를 매핑할 수 있도록 제어 또는 해석할 수 있는 회로가 필요하다.

<17> 도 1은 종래의 성좌 매핑 장치의 블록도이다.

<18> 종래의 성좌 매핑 장치는 직-병렬 변환부(Serial to Parallel Converter)(110), 주소 계산부(Address Computation Block)(120), 그리고 룩업 테이블이 저장된 롬(ROM)(130)으로 구성되어 있다.

<19> 직-병렬 변환부(Serial to Parallel Converter)(110)는 직렬 데이터를 입력받아 병렬 데이터로 변환시켜 주소 계산부(Address Computation Block)(120)로 전달한다.

<20> 주소 계산부(Address Computation Block)(120)는 비트 당 심벌(Bits-per-Symbol) 정보와 입력 데이터를 이용하여 해당되는 성좌의 주소를 발생시키는데, 이 주소 정보를 룩업 테이블(Look-up Table)의 형태로 만들어 출력할 수 있다.

<21> 이때 비트 당 심벌(Bits-per-Symbol) 정보를 가지고 여러 개의 성좌, 즉 n -

QAM(여기서 n 은 4, 8, 16, 32, ..., 2^n (입력되는 bit수))중에서 하나를 선택한다. 예를 들어 4-QAM에서 256-QAM까지의 성좌도를 처리할 경우, 비트 당 심벌(Bits-per-Symbol)은 3비트로 나타낼 수 있다. 4-QAM에서 256-QAM까지, 모두 7개의 성좌가 있으므로 비트 당 심벌(Bits-per-Symbol) 정보는 3비트로 나타낼 수 있고, 만일 그 값이 '001'일 경우 4-QAM을, '111'일 경우 256-QAM을 선택한다.

<22> 그리고, 롬(ROM)(130)은 상술한 바에 따라 계산된 성좌의 주소정보들이 기록된 록업 테이블을 저장하고 있다.

<23> 도 2는 4-QAM과 16-QAM의 성좌 매핑도를 나타낸 도면이다.

<24> 도 2를 참조하면, 4-QAM과 16-QAM을 처리해야 하는 성좌 매핑 장치일 경우, 비트 당 심벌(Bits-per-Symbol)정보는 1 비트로 표시할 수 있고('0' 인 경우 4-QAM이고, '1' 인 경우 16-QAM), 병렬로 변환된 입력 데이터는 4 비트로 표시할 수 있다. 그리고 성좌 점을 저장하는 롬(ROM)은, 4-QAM의 경우 4개의 성좌 점이 그리고 16-QAM의 경우 16개의 성좌 점이 존재하므로, 인-페이즈(In-phase) 및 Quadrature 각 성분에 대해 모두 20개의 성좌 점을 표시해야 하기 때문에 20개의 워드(word)를 저장한다.

<25> 도 3은 성좌 매핑을 위한 메모리 구성을 나타낸 도면이다.

<26> 도 3을 참조하면, 주소 계산부는 병렬 입력 데이터와 비트 당 심벌(Bits-per-Symbol) 정보를 이용하여 비트 당 심벌(Bits-per-Symbol)정보가 '0' 일 때는 4-QAM의 주소를, '1'일 때는 16-QAM의 주소를 발생시킨다. 따라서, 4-QAM에서 256-QAM까지의 성좌를 처리할 경우, 메모리의 1 워드(word)를 8 비트(bits)로 표시한다면, 1

Kbyte(512Byte x 2) 크기의 롬(ROM)이 필요하게 되어, 그 크기가 무시하지 못할 정도로 커지게 된다.

【발명이 이루고자 하는 기술적 과제】

<27> 본 발명이 이루고자 하는 기술적 과제는, 각각의 성좌 점 전체를 저장하는 메모리를 이용하지 않고 각 성좌도의 사분면 중 하나의 사분면상에 있는 성좌 점들만을 저장하여 메모리의 크기를 1/4로 줄일 수 있고, 이러한 메모리를 이용하기 위한 주소를 발생시키는 성좌 매핑 장치를 제공하는데 있다.

<28> 상술한 바와 같이 종래의 성좌 매핑 장치(Constellation Mapper)는 채널 상황에 따라 전송 속도를 달리 하는 고속 데이터 통신을 하는 경우에는 여러 개의 성좌도를 처리해야 하기 때문에, 메모리와 주소 생성 회로가 커질 수 밖에 없다. 따라서, 본 발명에서는 이러한 문제점을 해결하기 위해 여러 개의 성좌도에서 모든 성좌 점들을 메모리에 저장하지 않고 각 성좌도의 1사분면 내의 성좌 점들만 저장하여 필요한 메모리의 크기를 1/4로 줄일 수 있는 성좌 매핑 장치를 제공한다.

【발명의 구성 및 작용】

<29> 상기의 과제를 이루기 위하여 본 발명에 의한 성좌 매핑 장치는, 각 성좌의 사분면 중 어느 한 분면의 성좌값을 저장하고 있는 메모리; 성좌 점 데이터, 비트 당 심벌(Bits-per-Symbol) 정보, 그리고 상기 비트 당 심벌 정보가 유효한지 아닌지를 나타내는 심벌 유효 정보를 입력받아 상기 성좌 점 데이터에 해당하는 성좌값이 저장되어 있는 상기 메모리의 주소정보 및 상기 성좌 점 데이터가 위치하고 있

는 사분면을 나타내는 사분면 정보를 발생하는 주소 생성부; 상기 주소 생성부로부터 입력받은 상기 성좌 점 데이터에 대한 상기 사분면 정보를 기초로 상기 주소정보에 의해 상기 메모리로부터 읽은 상기 성좌값을 반전 또는 비반전시키는 반전 회로부; 및 상기 비트 당 심벌정보의 전송속도를 나타내는 보드율을 기초로 소정의 이득을 상기 반전 회로부의 출력에 곱하여 출력하거나, 상기 반전 회로부의 출력을 그대로 출력하는 스케일러부를 구비한다.

<30> 상기의 과제를 이루기 위하여 본 발명에 의한 성좌 매핑 방법은, 성좌 점 데이터, 비트 당 심벌(Bits-per-Symbol) 정보, 그리고 상기 비트 당 심벌 정보가 유효한지 아닌지를 나타내는 심벌 유효 정보를 입력받는 단계; 상기 입력받은 성좌 점에 대한 성좌값을 저장하고 있는 메모리의 주소 정보를 생성하는 단계; 상기 생성된 메모리의 주소정보를 기초로 성좌의 사분면 중 어느 한 분면에서의 실수값 및 허수값을 발생시켜 메모리의 출력을 그대로 또는 반전시켜 출력하는 단계; 및 상기 출력값에 소정의 이득을 곱한 값 또는 상기 출력값 중에서 하나를 선택하여 출력하는 단계를 구비한다.

<31> 상기한 과제를 이루기 위하여 본 발명에서는, 상기 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

<32> 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 일실시예를 상세히 설명한다.

<33> 도 4A 내지 도 4G는 본 발명에서 처리하는 성좌의 종류를 나타낸 도면이다.

<34> 본 발명의 구성에 대한 설명에 앞서, 도 4A 내지 도 4G를 참조하여 본 발명

이 처리하는 성좌의 종류와 값을 먼저 설명한다. 도 4A 내지 도 4G에서 보는 바와 같이 4-QAM(QPSK)에서 256-QAM 까지 7개의 성좌를 처리하고자 하며, 각 성좌에서의 각 성좌 점의 값은 임의의 값으로 정할 수 있으나 본 발명에서는 설명의 편의상 이진수 입력값에 해당되는 십진수 값을 성좌 점의 값으로 한다.

<35> 도 5는 본 발명의 성좌 매핑 장치의 간략한 구성을 나타낸 블록도이다.

<36> 본 발명의 성좌 매핑 장치는 주소 생성부(Address Generation block)(510), 메모리(520), 반전 회로부(Complement Logic Block)(530), 스케일러부(Scaler Block)(540) 및 래치부(Latch Block)(550)로 구성된다.

<37> 주소 생성부(Address Generation block)(510)는 성좌 점 데이터(511), 비트 당 심벌(Bits-per-Symbol) 정보(512), 그리고 입력 데이터가 유효한지 아닌지를 나타내는 유효 심벌 정보(valid_symbol)(513)를 입력받아 성좌 점에 대한 주소를 발생시킨다.

<38> 메모리(520)는 각 성좌의 사분면 중 하나의 사분면 상의 성좌 값을 저장하고 있으며, 이는 롬(ROM)이 될 수 있다.

<39> 반전 회로부(Complement Logic Block)(530)는 상기 롬(ROM) 또는 메모리에 저장된 성좌점들이 사분면 중 어느 사분면인가를 알려주는 신호인 quad_info 신호에 따라 메모리(520)로부터 전송되어 오는 데이터를 반전(complement) 또는 비반전시키는 기능을 수행한다.

<40> 스케일러부(Scaler Block)(540)는 반전 회로부(Complement Logic Block)(530)의 출력을 입력받아 입력 보드 율(Baud Rate)(541)에 따라 이득을 선별적으로 곱하여 출력한다.

- <41> 그리고 래치부(Latch Block)(550)는 스케일러부(Scaler Block)(540)의 출력을 다음에 연결된 블록을 위해 재타이밍(re-timing) 시켜주는 기능을 수행한다.
- <42> 도 6은 본 발명의 성좌 매핑 장치의 상세한 구성을 나타낸 도면이다.
- <43> 주소 생성부(Address Generation block)(610)는 일반적인 마이크로 컨트롤러에서의 인덱스트 어드레싱(indexed addressing) 방식을 사용한다. 인덱스트 어드레싱(indexed addressing) 방식은, 실제 유효한 어드레스의 값은 베이스 어드레스(base address)의 값에 인덱스 어드레스(index address)의 값을 더한 값이 되는 어드레스 지정 방식을 말한다. 따라서 주소 생성부(610)는 베이스 어드레스(base-address)를 생성하는 베이스 어드레스 생성부(611), 인덱스 어드레스(index-address)를 생성하는 인덱스 어드레스 생성부(614), 그리고 베이스 어드레스(base-address)와 인덱스 어드레스(index-address)를 더하는 덧셈기(618)로 구성되어 있다. 그러므로, 예를 들어 베이스 어드레스(base address)가 3번지를 가리키고 인덱스 어드레스(index address)가 5번지를 가리키면 실제 어드레스(address)는 $3+5=8$ 이 되어 8번지가 유효번지가 된다.
- <44> 그리고, 베이스 어드레스 생성부(611)는 멀티플렉서(612)와 쉬프트기(613)로 구성되어 있다.
- <45> 멀티플렉서(612)는 입력 비트 당 심벌(Bits-per-Symbol) 정보와 유효 심벌 정보(valid-symbol)를 입력받아, 유효 심벌 정보(valid-symbol)의 값에 따라 비트 당 심벌(Bits-per-Symbol) 값 또는 0 의 값중에서 하나를 선택한다.
- <46> 쉬프트기(613)는 '00000001'의 값을 멀티플렉서(612)의 출력 값만큼 이동시킨다.

- <47> 그리고, 인덱스 어드레스 생성부(614)는 감산기(615), 쉬프트기(616) 및 멀티플렉서(617)로 구성되어 있다.
- <48> 감산기(615)는 베이스 어드레스 생성부(611)내의 멀티플렉서(612)의 출력값에서 이진값 '111'을 빼는 기능을 수행한다.
- <49> 쉬프트기(616)는 입력되는 8비트의 데이터를 감산기(615)의 출력값만큼 이동시키는 기능을 수행한다.
- <50> 멀티플렉서(617)는, 쉬프트기(616)의 출력 중 하위 6 비트(0 ~ 5번째까지의 비트)를 선택하여 이중에서 최상위 비트, 즉 5번째 비트의 값과 이진값 '00000'을 연결시켜 구성한 신호(671)와, 쉬프트기(616)의 하위 6 비트의 출력에서 5 ~ 4번째 비트의 값과 이진값 '0000'을 연결시켜 구성한 신호(672)와, 5 ~ 3번째 비트의 값과 이진값 '000'을 연결시켜 구성한 신호(673)와, 5 ~ 2번째 비트의 값과 이진값 '00'을 연결시켜 구성한 신호(674)와, 5 ~ 1번째 비트의 값과 이진값 '0'을 연결시켜 구성한 신호(675)와, 이진값 '000000'을 입력 단자로 하고 베이스 어드레스 생성부(611) 내의 멀티플렉서(612)의 출력을 선택단자로 한다.
- <51> 메모리(620)는 각 성좌의 1 사분면 성좌 값을 저장하고 있으며, 이는 롬(ROM)이 될 수 있다. 롬(ROM)은 실수 성분을 저장하고 있는 REAL ROM(621)과 허수 성분을 저장하고 있는 IMAGINARY ROM(622)로 구성되어 있다.
- <52> 반전 회로부(Complement Logic Block)(630)는 메모리(620)의 인페이즈(In-phase) 출력과 Quadrature-phase의 출력을 입력받는다.

- <53> 그리고, 반전부(631)(632)는 메모리(620)에 저장된 성좌점들이 각 성좌의 어느 사분면에 위치하고 있는지 알려 주는 quad_info신호를 입력으로 하고, I 데이터 및 Q 데이터 각각의 입력에 대해 음의 값을 만든다.
- <54> 반전 제어부(635)는 quad_info 입력과 인덱스 어드레스 생성부(614)의 쉬프트기(616) 출력 중 상위 2비트(quad_real, quad_imag)를 입력으로 하여, 메모리에 저장된 성좌점이 사분면 중 어느 분면의 성좌점 인가에 따라 메모리의 각 출력을 반전시키거나 또는 비반전시키는 제어 신호를 발생시킨다.
- <55> 그리고, 멀티플렉서(633)(634)는 반전 제어부(635)의 출력신호를 선택 단자로 입력 받는다.
- <56> 스케일러부(Scaler Block)(640)는 인페이즈(In-phase) 신호와 Quadrature-phase 신호에 대한 스케일링(scaling) 두 부분으로 나누어져 있다. 인페이즈(In-phase)에 대한 스케일러(scaler)는 인덱스 어드레스 생성부(614)의 쉬프트기(616) 출력 중 일곱번째 비트의 데이터를 멀티플렉서(641)의 선택단자로 입력받는다. 그리고 이 멀티플렉서(641)는, 반전 회로부(Complement Logic Block)(630)의 인페이즈(in-phase) 출력과 그 출력을 보드율(baud-rate) 값에 따라 이득(K)(642)을 곱한 것을 입력으로 한다.
- <57> Quadrature-phase에 대한 스케일러(scaler)는 멀티플렉서(643)의 선택단자가 인덱스 어드레스 생성부(614)의 쉬프트기(616) 출력 중 여덟번째 비트라는 것을 제외하고는 상술한 인페이즈(in-phase)의 경우와 동일하다.

- <58> 그리고 래치부(Latch Block)(650)는 스케일러부(Scaler Block)(640)의 출력을 다음에 연결된 블록을 위해 재타이밍(re-timing) 시켜주는 기능을 수행한다. 래치부(650)는 2개의 플립플롭(Flip Flop, FF)(651)(652)으로 구성되어 있으며, 스케일러부(640)에서의 멀티플렉서(641)(643)의 출력을 입력받는다.
- <59> 본 발명의 성좌 매핑 장치의 입력신호는 8 비트의 입력 데이터 신호, 심벌 당 매핑되어야 할 비트 수를 알려주는 3비트의 비트 당 심벌(Bits-per-Symbol) 정보, 현재 입력되는 데이터가 유효한가를 알려주는 심벌 유효 정보(valid_symbol), 그리고 성좌 매핑 장치에서 심벌 또는 보드 속도(Symbol or Baud Rate)를 알려주는 입력 보드율(baud_rate)이 있다. 그리고, 출력신호는 각 심벌의 인페이즈(In-phase) 및 Qaudrature-phase에 해당되는 ConsMap_I, ConsMap_Q 가 있다.
- <60> 입력 데이터가 8비트가 되는 것은 본 발명의 성좌 매핑 장치가 256-QAM까지 매핑을 할 수 있도록 위해서이다. 그리고, 비트 당 심벌(Bit-per-Symbol) 값에 따라서 입력으로 선택되는 또는 데이터로 인식되는 비트 수가 달라진다. 즉, QPSK(4-QAM)일 경우, 비트 당 심벌(Bits-per-Symbol)은 '001'이 되고, 입력 데이터는 하위 두 비트만 유효 데이터로 채워진다. 입력 데이터의 0 ~ 1번째 비트에만 유효한 데이터가 채워지므로 나머지 비트는 유효하지 않은 데이터가 된다. 또, 8-PSK(8-QAM)일 때는 비트 당 심벌(Bits-per-Symbol)은 '010'이 되고, 입력 데이터는 0 ~ 2번째 비트까지 유효한 입력이 된다. 이렇게 해서, 256-QAM일 경우 비트 당 심벌(Bits-per-Symbol)은 '111'이 되고, 유효 입력 데이터의 비트는 0 ~ 7까지가 된다.
- <61> 도 7은 비트 당 심벌값에 따른 유효 입력 데이터를 나타낸 도표이다.

<62> 도 7에서 나타낸 바와 같이 입력 8비트 데이터는 비트 당 심벌(Bits-per-Symbol)값에 따라 유효 입력 비트수가 다르고, 유효 입력 비트 중 상위 2비트는 도 4A 내지 도 4G에서 각 성좌 점들이 속한 사분면의 위치의 파악시 이용할 수 있다. 즉, 도 4A 내지 도 4G에서의 모든 성좌 점들에 대응되는 비트를 보면 상위 두 비트는 각 사분면에 대해서 동일하다. 1사분면 상의 점들은 '00', 2사분면은 '01', 3사분면은 '11', 4사분면은 '10'이라는 것을 알 수 있다.

<63> 이와 같이 입력된 데이터에서 상위 두 비트를 각 점에 대한 사분면을 나타내는 정보로 이용할 수 있다. 따라서, 모든 입력에 대응되는 성좌 점들을 메모리에 저장하지 않고, 사분면 중 어느 한 분면(즉, 1, 2, 3, 4분면 중 어느 한 분면)에 있는 점들만 저장하고 나머지 점들에 대해서는 저장된 성좌점이 어느 사분면상의 점들인가를 알 수 있는 정보와 어드레스 발생부에서 발생시킨 `quad_real`, `quad_imag` 신호를 이용하여 어느 사분면에 있는지를 알아내어 처리할 수 있다.

<64> 예를 들어, 도 4C의 16-QAM 성좌도 내의 '0110'에 해당되는 성좌 점의 경우를 보면, 상위 두 비트 '01'은 '0110'에 해당되는 성좌 점이 2사분면에 속하고 1사분면에 있는 '0010'의 점과 허수축으로부터 같은 거리에 있다는 것을 알 수 있다. 이는 '0010'의 성좌 점에 해당되는 심벌값에서 실수값의 부호만 반대이고 크기는 동일하다는 것이다.

<65> 만약 메모리에 1사분면 상의 점들이 저장되어 있다면 나머지 점들은 이들로부터 구할 수 있다. 즉 2사분면 상의 점들은 1사분면 상의 점들로부터 실수축 값을 반전하면 되고, 3사분면 상의 점들은 실수축과 허수축의 값을 반전시키고, 4사분면 상에 있는 점들은 허수축의 값을 반전시키면 된다. 이런 식으로 메모리에 저장된 성좌점들이 2사분면이

나 3사분면 또는 4사분면 상의 점들을 저장한다 하더라도 위와 같은 방법으로 저장되지 않은 사분면들의 점들의 값을 구해 낼 수 있다. 따라서, 성좌 점에 해당되는 심벌의 값은 각 성좌에서 사분면 중 하나의 사분면에 해당되는 값만 메모리에 저장하고 저장된 성좌 점이 어느 사분면의 점들을 저장했는가를 알면 저장되어 있는 성좌 점들로부터 나머지 성좌 점들에 대한 심벌값을 매핑시킬 수 있다.

<66> 성좌 점에 해당되는 심벌의 값은 각 성좌에서 1사분면의 값만 메모리에 저장되고, 성좌 점이 어느 사분면에 있는가를 알아내어 1사분면에 속한 성좌 점의 심벌 값에 대해서 실수 또는 허수축의 값의 부호를 반전시키면 된다는 것을 알 수 있다.

<67> 도 8은 본 발명의 성좌 매핑 장치에서 사용되는 메모리의 구성도를 나타낸 도면이다.

<68> 도 8을 참조하여 성좌 정보를 메모리에 저장하는 방법을 설명한다. 여기에서는 예를 들어 1사분면 내의 성좌 점들이 저장되어 있는 경우를 설명한다. 도 8에서 보듯이 도 4A 내지 도 4G에 나와 있는 각 성좌의 1사분면 내의 성좌 점들이 메모리에 저장된다. 인페이즈(In-Phase) 성분의 성좌 메모리(REAL ROM)(621)의 경우에 저장되는 순서를 보면, 먼저 성좌 매핑 장치의 출력이 0이 되게 하는 제로 심벌(Zero Symbol)을 0번지에 저장하고, 그 다음 번지에는 4-QAM(QPSK)의 1사분면 내의 성좌 점(도 4A에서 0번째 성좌 점에 해당됨)의 실수값이 저장되고, 2 ~ 3번지에는 8-QAM(8-PSK) 성좌의 1사분면 내의 성좌 점(도 4B에서 0, 1번째 점)의 실수값이 저장된다. 계속해서 이런 방식에 의하여 256-QAM의 경우 65 ~ 127번지에 256-QAM성좌의 1사분면 내의 점들의 실수값이 저장된다.

- <69> Quadrature 성분의 메모리(IMAGINARY ROM)(622)의 경우 메모리에 저장된 값이 실수 값이 아니고 허수값이라는 것을 제외하고는 인페이즈(In-phase) 성분의 메모리의 경우와 같다.
- <70> 먼저 주소 생성부(610)가 베이스 어드레스(base-address)를 만드는 것부터 설명한다. 입력 비트 당 심벌(Bits-per-symbol)을 가지고 상술한 바와 같이 현재 입력되고 있는 데이터가 어느 성좌인지를 알 수 있다. 따라서, 비트 당 심벌(Bits-per-symbol) 정보로부터 해당하는 성좌의 베이스 어드레스(base-address)를 찾아 낼 수 있다.
- <71> 메모리의 각 성좌에 해당되는 베이스 어드레스(base-address)는 0, 1, 2, 4, 8, ..., 65번지이고 비트 당 심벌(Bits-per-symbol) 값에 따라, 비트 당 심벌 값에 해당하는 시작 번지를 발생시켜야 한다. 현재 입력되는 데이터가 유효하지 않은 심벌이라면 베이스 어드레스 생성부(611)의 멀티플렉서(612)는 0의 값을 출력하고 그렇지 않은 경우 비트 당 심벌(Bits-per-Symbol) 값을 출력한다. 즉, 심벌유효정보(valid_symbol)가 0이면 멀티플렉서(612)의 출력이 0이 되고, 따라서 '00000001'을 멀티플렉서(612)의 출력 값만큼 이동시키는 쉬프터기(613)와 그 이동된 값 중에서 상위 7 비트만을 선택하는 선택기에 의해 베이스 어드레스(base-address) 0이 된다.
- <72> 심벌유효정보(valid_symbol)가 1인 경우 '00000001'의 값을 비트 당 심벌(Bits-per-Symbol) 값만큼 이동시킨다. 비트 당 심벌(Bits-per-Symbol) 값이 이진 값으로 '010' 이라면 8-QAM을 나타내고, 이 값만큼 '00000001'을 좌측으로 2번 이동시켜 '00000100'이 되고, 이동시킨 값 중 상위 7비트만 선택하면 '0000010'이 되어 베이스 어드레스(base-address)는 2가 된다.

- <73> 이 베이스 어드레스(base address)가 성좌 점들이 저장된 메모리에서 8-QAM의 시작 번지가 되는 것이다. 이렇게 하여 메모리의 베이스 어드레스(base-address)를 발생시키고, 그리고 나서 입력 데이터로부터는 비트 당 심벌(Bits-per-Symbol)값에 의해 선택된 성좌내의 성좌점 들을 매핑하는 인덱스 어드레스(index-address)를 발생시켜야 한다. 도 7을 참조하면, 입력 8비트 데이터는 비트 당 심벌(Bits-per-Symbol) 정보에 따라 유효 입력 비트수가 달라지고 유효 입력 비트 중 상위 2비트는 도 4A 내지 도 4G에서 나타내듯이 각 성좌 점들에 대한 사분면의 위치를 파악하는데 이용할 수 있다.
- <74> 즉, 도 4A 내지 도 4G에서의 모든 성좌 점들에 대응되는 비트는 동일한 사분면인 경우에 상위 두 비트가 동일하다. 즉, 1사분면 상의 점들은 '00', 2사분면 상의 점들은 '01', 3사분면 상의 점들은 '11', 4사분면 상의 점들은 '10' 이라는 것을 알 수 있다.
- <75> 이와 같이 입력 데이터로부터, 데이터가 매핑될 성좌 점의 사분면을 나타내는 상위 두 비트와 선택된 성좌에서 성좌 점에 매핑될 나머지 비트를 분리하는 것이 필요하다. 유효 입력 데이터의 상위 두 비트는 각 성좌마다 위치가 달라 성좌 마다 다른 방법으로 찾아내야 한다. 이러한 것을 방지하기 위해서 상위 두 비트의 위치가 선택된 성좌에 관계없이 동일한 위치에 존재하도록 입력된 데이터를 이동시키는데, 7에서 감산기(615)에서의 출력값(Shfnum)을 뺀 값만큼 이동시킨다.
- <76> 도 9는 입력 데이터를 이동시킨 후의 값을 나타낸 도면이다.
- <77> 그리고, 도 8을 참조하면, 선택된 성좌에 관계없이 상위 두 비트가 사분면을 나타낼 수 있다. 이들 두 비트를 제외한 나머지 여섯 비트(0 ~ 5번째 데이터)는 데이터가 속한 사분면 내에서의 성좌 점에 대응되도록 인덱스 어드레스(index-address)를 발생

시키는데 이용된다. 이들 여섯 비트를 각 성좌에 해당되는 비트 수 만큼 선택한 신호를 분리해 내어 비트 당 심벌(Bits-per-symbol) 값에 따라 선택할 수 있게 한다.

<78> 즉, 8-QAM인 경우 6비트 중 실제 의미 있는 비트는 5번째 비트 1개이고, 16-QAM은 5 ~ 4번째 비트 2개, 32-QAM은 5 ~ 3번째 비트 3개, 64-QAM은 4개, 126-QAM 은 5개, 256-QAM 은 6개가 된다. 그리고 4-QAM일 때는 유효 비트 중 사분면을 나타내는 비트 수가 2개이므로 인덱스 어드레스(index-address)를 발생시키는 6비트에는 포함되지 않는다. 이렇게 선택된 신호들은 비트 당 심벌(Bits-per-Symbol)을 선택 단자로 하는 멀티플렉서(617)의 2, 3, 4, 5, 6, 7 번째입력단자로 연결되고 4-QAM이나 유효하지 않은 데이터(valid_symbol 값이 0인 경우)일 경우에는 0의 값이 멀티플렉서(617)의 0, 1번째 입력단자로 연결된다.

<79> 이렇게 하여 만들어진 인덱스 어드레스(index-address)는 베이스 어드레스(base-address)와 더해져서 성좌 점에 해당되는 심벌 값을 저장한 메모리의 어드레스를 생성시키는 것이다.

<80> 예를 들어, 비트 당 심벌(Bits-per-Symbol) 값이 '011', 즉, 3 이고, 심벌유효정보(valid-symbol)가 1이고, 입력 데이터가 "00001101"인 경우를 살펴보자. 비트 당 심벌(Bits-per-Symbol) 값이 3 이므로 16-QAM을 나타낸다. 이에 대한 베이스 어드레스(base-address)는 "00000001"의 값을 좌측으로 3번 이동시킨 후 상위 7비트의 값을 선택한 것이므로 "0000100"이 되어 메모리의 4번지를 가리킨다.

<81> 인덱스 어드레스(Index-address)는 입력 데이터 "00001101"을 $7 - \text{Shfnum} = 4$ ($\text{Shfnum} = 3$ 이므로)만큼 좌측로 이동하면 '11010000'이 되고, 이중에서 상위 두 비트 '11'은 사분면 중 3사분면을 나타내고, 나머지 여섯 비트는 '010000'가 된다. 이중 의미

있는 값은 5 ~ 4번 째 비트 '01'이 되고 이를 다시 '0000'과 연결시켜 '000001'을 만든다. 이 '000001' 값은 멀티플렉서의 선택 단자에 의해 출력되어 덧셈기(618)로 들어간다. 따라서, 최종적으로 주소 생성부(610)의 출력은 $\text{base-address} + \text{index address} = '0000100' + '000001'$ 이 되어 $4 + 1 = 5$ 번지가 된다.

<82> 이렇게 발생된 메모리의 주소는 성좌의 사분면 중 한 분면에서의 실수 및 허수 값을 발생시키게 된다. 메모리의 출력은 다시 반전 회로부(Complement Logic Block)로 입력되어 선택된 성좌 점이 속한 사분면의 값이 발생될 수 있도록 한다.

<83> 반전 회로부(Complement Logic Block)는 입력 quadrant_info 신호를 가지고 현재 메모리에 저장된 성좌 점들이 사분면 중 어느 사분면에 있는가를 알 수 있다. 예를 들어 quadrant_info가 이진값 "00" 이면 1사분면, "01"이면 2사분면, "11"이면 3사분면, "10"이면 4사분면에 위치하고 있음을 나타낸다. 이들 정보와 함께 quad_real, quad_imag의 값을 반전 회로부(Complement Logic Block)에서 처리하여 메모리의 출력 또는 반전된 출력을 선택할 수 있는 멀티플렉서의 선택단자에 전송한다.

<84> 메모리에 저장된 성좌점들이 어느 분면의 점들인가에 따라 반전 및 비반전을 선택하는 것이 달라진다. 즉, 입력되는 데이터가 3사분면 상의 점이고 메모리에 저장된 성좌 값이 2사분면 상의 점들이라면, 실수축의 값은 반전되지 않은 값을, 허수축의 값은 반전된 값을 선택해야 한다. 그렇지만 메모리에 저장된 값이 4사분면 상의 점들이라면, 실수축의 값은 반전된 값을, 허수축의 값은 비반전된 값을 선택해야 한다.

<85> 입력데이터가 매핑되어야 할 성좌점의 분면과 메모리에 저장된 분면에 따라 반전 또는 비반전의 선택신호를 발생시키는 역할을 하는 것이 반전 제어부(635)이다. 입력신호 quadrant_info 신호는 메모리에 저장된 성좌점이 사분면 중 사분면에 있음을 나타내

고, quad_real, quad_imag 값은 입력된 데이터가 매핑되어야 할 사분면 중 하나를 나타내기 때문에 이를 이용하여 반전 및 비반전의 선택신호를 발생 시킬 수 있다.

- <86> 도 10A 내지 도 10D는 반전 및 비반전의 선택신호의 발생에 관한 진리표와 논리식을 나타낸 도면이다.
- <87> 도 10A는 저장된 성좌점이 1사분면상의 점일 경우(quadrant_info = "00")를, 도 10B는 저장된 성좌점이 2사분면상의 점일 경우(quadrant_info = "01")를, 도 10C는 저장된 성좌점이 3사분면상의 점일 경우(quadrant_info = "11")를, 그리고 도 10D는 저장된 성좌점이 4사분면상의 점일 경우(quadrant_info = "10")를 나타낸다. 이때 q1은 quad_imag를, q0는 quad_real을 나타내고, S1은 허수축의 멀티플렉서 선택 단자를, 그리고 S2는 실수축의 멀티플렉서 선택 단자를 나타낸다.
- <88> 도 10B를 참조하면, 입력 신호 quad_real(qr0)과 quad_imag(qr1)은 입력된 신호에 대해 어드레스 생성부를 거쳐 나온 데이터의 사분면 위치를 알려주는 신호이다. 여기서 "00"은 1사분면임을 나타내고, "01"은 2사분면임을, "11"은 3사분면임을, 그리고 "10"은 4사분면임을 나타내는 것으로 가정한다, 도 10B는 저장된 성좌점들이 2사분면에 있는 점들이므로, $(qr0, qi1) = (0, 0)$ 일때 즉, 입력 데이터가 1사분면 상의 점일 때 반전 회로부는, 메모리로부터의 실수 출력 I를 반전시키기 위한 신호를 선택하여야 하고 허수출력 Q는 비반전시켜 그대로 출력되도록 하는 신호를 선택해야 한다.
- <89> 따라서, $(qr0, qi1) = (0, 0)$ 일 경우 선택단자 (S1, S0)은 (+, -) (여기서, +는 비반전된 신호를, -는 반전된 신호를 선택한다는 의미)를 출력하도록 한다. 이와 같은 방법을 다른 사분면인 경우에 대해서도 적용할 수 있다.

- <90> 도 11은 반전 제어부(635)를 나타낸 도면이다.
- <91> 도 11을 참조하면, quadrant_info에 따라 데이터의 사분면 중 한 분면을 알려주는 quad_real, quad_imag를 선택하여 메모리의 I 출력 및 Q 출력에 대해 반전 또는 비반전된 출력을 선택할 수 있는 신호를 발생시킴을 알 수 있다.
- <92> 도 12는 본 발명의 성좌 매핑 방법을 설명한 플로우차트이다.
- <93> 우선, 비트 당 심벌(Bits-per-Symbol) 값, 심벌유효정보(valid-symbol) 및 입력 데이터를 주소 생성부(510)로 입력받는다(S1210). 입력받은 데이터를 가지고 상술한 방법에 의해 메모리의 주소 정보를 생성한다(S1220).
- <94> 이렇게 생성된 메모리의 주소는 성좌의 사분면 중 어느 한 분면에서의 실수값 및 허수값을 발생시킨다(S1230). 즉, 메모리의 출력은 반전 회로부(Complement Logic Block)(530)로 입력되어 선택된 성좌 점이 속한 사분면의 값이 발생될 수 있도록 한다. 반전 회로부(Complement Logic Block)는 앞서 설명한 quadrant_info 정보를 통해 현재 메모리에 저장된 성좌 점들이 사분면 중 어느 사분면에 있는 것인가를 알 수 있다. 이들 정보와 함께 quad_real, quad_imag의 값을 반전 제어부(635)에서 처리하여 메모리의 출력 또는 반전된 출력을 선택해서 출력할 수 있도록 한다.
- <95> 그리고, 상기 반전 회로부(Complement Logic Block)(530)의 출력에 소정의 이득을 곱한 값 또는 반전 회로부의 출력을 선택하여 출력한다(S1240). 즉, 인페이즈(in-phase) 출력과 그 출력을 보드율(baud-rate) 값에 따라 이득(K)을 곱한 정보를 입력받는다. 이와 함께 Quadrature-phase 출력에 대해서도 동일하게 상술한 과정이 적용된다. 다만 인페이즈(In-phase) 신호의 경우에는 상기 인덱스 어드레스 생성부의 쉬프트기 출력 중 일

곱번째 비트의 데이터를 멀티플렉서의 선택단자로 입력받는데에 비하여,
 Quadrature-phase 신호의 경우에는 여덟번째 비트의 데이터를 멀티플렉서의 선택단자로
 입력받는다.

<96> 마지막으로, 래치부(Latch Block)(550)가 선택하여 출력된 신호를 저장하거나 다음
 에 연결된 블록을 위해 재타이밍(re-timing)을 시킨다(S1250).

<97> 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로
 서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여
 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을
 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터
 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구
 현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨
 터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수
 있다.

<98> 이제까지 본 발명에 대하여 그 바람직한 실시예들을 중심으로 살펴보았다. 본 발
 명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특
 성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을
 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되
 어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그
 와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이
 다.

【발명의 효과】

- <99> 상술한 바와 같이 본 발명은, 여러 개의 성좌도에서 모든 성좌 점들을 메모리에 저장하지 않고 각 성좌도의 사분면 중 하나의 사분면 내의 성좌 점들만 저장함으로써, 요구되는 메모리의 크기를 1/4로 줄일 수 있는 효과가 있다.
- <100> 또한, 각 성좌도의 점들에 대한 어드레스 발생 부분은 각 성좌도에 대한 시작 어드레스를 발생시키는 부분과 입력 데이터를 처리하여 비트 당 심벌(Bits-per-Symbol)에 해당되는 성좌도내 성좌 점에 대한 어드레스를 발생시키는 부분으로 나누어 처리해 성좌가 많아 질 경우 룩업 테이블(Look-up table)이나 롬(ROM)을 사용하는 것에 비해 크기를 줄일 수 있는 효과가 있다.
- <101> 그리고, 최근에 고속의 통신을 위해 처리해야 될 성좌가 많은 홈 PNA(HomePNA)와 같은 맥내 데이터 통신에서, 변조시 사용되는 성좌 처리기의 크기를 줄임으로써, 변조기 전체의 하드웨어 복잡도를 줄일 수 있는 효과가 있다. 따라서, 본 발명의 성좌 매핑 장치를 사용하여, 다양한 성좌도를 처리해야 하는 고속 통신이나 홈 PNA(HomePNA)와 같은 맥내 데이터 통신 시스템에서 변조시 사용되는 성좌 처리기의 크기를 줄임으로써, 변조기 전체의 하드웨어 복잡도를 줄이고, 전체 모뎀 칩의 면적을 줄일 수 있는 효과가 있다.

【특허청구범위】**【청구항 1】**

각 성좌의 사분면 중 어느 한 분면의 성좌값을 저장하고 있는 메모리;

성좌 점 데이터, 비트 당 심벌(Bits-per-Symbol) 정보, 그리고 상기 비트 당 심벌 정보가 유효한지 아닌지를 나타내는 심벌 유효 정보를 입력받아 상기 성좌 점 데이터에 해당하는 성좌값이 저장되어 있는 상기 메모리의 주소정보 및 상기 성좌 점 데이터가 위치하고 있는 사분면을 나타내는 사분면 정보를 발생하는 주소 생성부;

상기 주소 생성부로부터 입력받은 상기 성좌 점 데이터에 대한 상기 사분면 정보를 기초로 상기 주소정보에 의해 상기 메모리로부터 읽은 상기 성좌값을 반전 또는 비반전시키는 반전 회로부; 및

상기 비트 당 심벌정보의 전송속도를 나타내는 보드율을 기초로 소정의 이득을 상기 반전 회로부의 출력에 곱하여 출력하거나, 상기 반전 회로부의 출력을 그대로 출력하는 스케일러부를 포함하는 것을 특징으로 하는 성좌 매핑 장치.

【청구항 2】

제1항에 있어서,

상기 스케일러부의 출력을 재타이밍시켜 출력하는 래치부를 더 포함하는 것을 특징으로 하는 성좌 매핑 장치.

【청구항 3】

제1항에 있어서, 상기 주소 생성부는

베이스 어드레스를 생성하는 베이스 어드레스 생성부;

인덱스 어드레스를 생성하는 인덱스 어드레스 생성부; 및
 상기 베이스 어드레스와 인덱스 어드레스를 더하는 덧셈기를 포함하는 것을 특징으로 하는 성과 매핑 장치.

【청구항 4】

제3항에 있어서, 상기 베이스 어드레스 생성부는
 입력 비트 당 심벌 정보와 심벌유효 정보를 입력받아, 상기 심벌유효 정보의 값에 따라 비트 당 심벌 정보 또는 0 의 값 중에서 하나를 선택하여 출력하는 멀티플렉서;
 및
 소정의 값을 상기 멀티플렉서의 출력값만큼 이동시키는 쉬프트기를 포함하는 것을 특징으로 하는 성과 매핑 장치.

【청구항 5】

제4항에 있어서, 상기 소정의 값은
 이진값 '00000001' 인 것을 특징으로 하는 성과 매핑 장치.

【청구항 6】

제3항에 있어서, 상기 인덱스 어드레스 생성부는
 상기 베이스 어드레스 생성부내의 상기 멀티플렉서의 출력값에서 소정의 값을 빼는 감산기;
 상기 입력되는 데이터를 상기 감산기의 출력값만큼 이동시키는 쉬프트기; 및
 상기 쉬프트기의 출력 중 하위 6 비트를 선택하여 이 중에서 최상위 비트의 값과 이진값 '00000'을 연결시켜 만든 데이터와,

상기 쉬프트기의 출력 중 하위 6 비트의 출력에서 5 내지 4번째 비트의 값과 이진 값 '0000'을 연결시켜 만든 데이터와,

상기 쉬프트기의 출력 중 하위 6 비트의 출력에서 5 내지 3번째 비트의 값과 이진 값 '000'을 연결시켜 만든 데이터와,

상기 쉬프트기의 출력 중 하위 6 비트의 출력에서 5 내지 2번째 비트의 값과 이진 값 '00'을 연결시켜 만든 데이터와,

상기 쉬프트기의 출력 중 하위 6 비트의 출력에서 5 내지 1번째 비트의 값과 이진 값 '0'을 연결시켜 만든 데이터와, 이진값 '000000'을 입력받고, 상기 베이스 어드레스 생성부 내의 멀티플렉서의 출력을 선택단자로 입력받는 멀티플렉서를 포함하는 것을 특징으로 하는 성좌 매핑 장치.

【청구항 7】

제1항에 있어서, 상기 메모리는

각 성좌의 사분면 중 어느 한 분면의 성좌 값을 저장하고 있으며 상기 성좌 값의 실수 성분과 허수성분을 분리하여 저장하고 있는 것을 특징으로 하는 성좌 매핑 장치.

【청구항 8】

제1항에 있어서, 상기 반전 회로부는

상기 메모리로부터의 인페이즈 출력과 Quadrature-phase의 출력을 입력받아 음의 값을 만드는 반전부;

상기 메모리로부터의 각 출력을 인덱스 어드레스 생성부의 쉬프트기 출력 중 상위 2비트를 선택단자로 입력받는 멀티플렉서; 및

상기 메모리에 저장된 성과점들이 어느 사분면 상의 점을 저장한 것인지 알려주는 정보와, 인덱스 어드레스 생성부의 쉬프트기 출력 중 상위 2비트 정보를 입력받아, 상기 메모리에 저장된 임의의 사분면 상의 점들이 위치한 사분면 정보와 입력 데이터가 위치한 사분면 정보에 따라 상기 메모리의 각 출력을 반전 또는 비반전시키는 선택하는 제어 신호 발생시키는 반전 제어부를 포함하는 것을 특징으로 하는 성과 매핑 장치.

【청구항 9】

제1항에 있어서, 상기 스케일러부는

상기 인덱스 어드레스 생성부의 쉬프트기 출력 중 일곱번째 비트의 데이터를 선택 단자로 입력받는 멀티플렉서를 포함하고 있으며,

상기 멀티플렉서는, 상기 반전 회로부의 인페이즈 출력과 그 출력에 보드올 정보에 따라 소정의 이득값을 곱한 값을 입력받는 것을 특징으로 하는 성과 매핑 장치.

【청구항 10】

성과 점 데이터, 비트 당 심벌(Bits-per-Symbol) 정보, 그리고 상기 비트 당 심벌 정보가 유효한지 아닌지를 나타내는 심벌 유효 정보를 입력받는 단계;

상기 입력받은 성과 점에 대한 성과값을 저장하고 있는 메모리의 주소 정보를 생성하는 단계;

상기 생성된 메모리의 주소정보를 기초로 성과의 사분면 중 어느 한 분면에서의 실수값 및 허수값을 발생시켜 메모리의 출력을 그대로 또는 반전시켜 출력하는 단계; 및

상기 출력값에 소정의 이득을 곱한 값 또는 상기 출력값 중에서 하나를 선택하여 출력하는 단계를 포함하는 것을 특징으로 하는 성과 매핑 방법.

【청구항 11】

제10항에 있어서,

상기 선택하여 출력된 신호를 저장하거나 다음에 연결된 블록을 위해 재타이밍을 수행하는 단계를 더 포함하는 것을 특징으로 하는 성좌 매핑 방법.

【청구항 12】

제10항에 있어서, 상기 주소 정보를 생성하는 단계는

베이스 어드레스를 생성하는 단계;

인덱스 어드레스를 생성하는 단계; 및

상기 생성된 베이스 어드레스와 상기 생성된 인덱스 어드레스를 더하는 단계를 포함하는 것을 특징으로 하는 성좌 매핑 방법.

【청구항 13】

제12항에 있어서, 상기 베이스 어드레스를 생성하는 단계는

입력 비트 당 심벌 정보와 심벌유효 정보를 입력받아, 상기 심벌유효 정보의 값에 따라 비트 당 심벌 정보 또는 0 의 값 중에서 하나를 선택하여 출력하고,

이진값 '00000001' 값을 상기 선택하여 출력한 값만큼 이동시켜 베이스 어드레스를 생성하는 것을 특징으로 하는 성좌 매핑 방법.

【청구항 14】

제12항에 있어서, 상기 인덱스 어드레스를 생성하는 단계는

상기 베이스 어드레스를 생성하는 단계에서 생성된 값에서 소정의 값을 빼는 단계

;

상기 입력되는 데이터를 상기 뺀 출력값만큼 이동시키는 단계; 및

상기 이동시킨 값 중에서 하위 6 비트를 선택하여 이중에서 최상위 비트의 값과 이진값 '00000'을 연결시켜 만든 데이터와,

상기 이동시킨 값 중에서 하위 6 비트의 출력에서 5 내지 4번째 비트의 값과 이진값 '0000'을 연결시켜 만든 데이터와,

상기 이동시킨 값 중에서 하위 6 비트의 출력에서 5 내지 3번째 비트의 값과 이진값 '000'을 연결시켜 만든 데이터와,

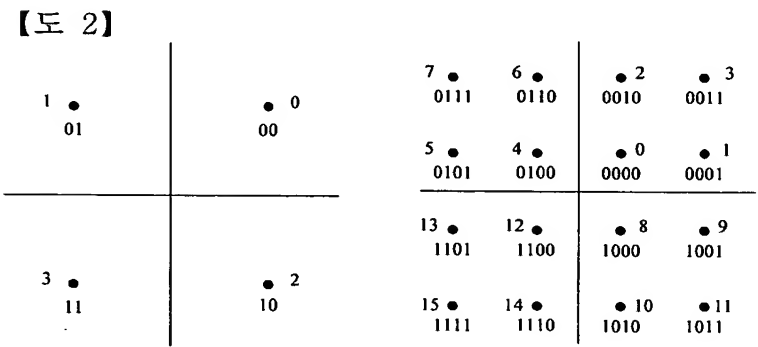
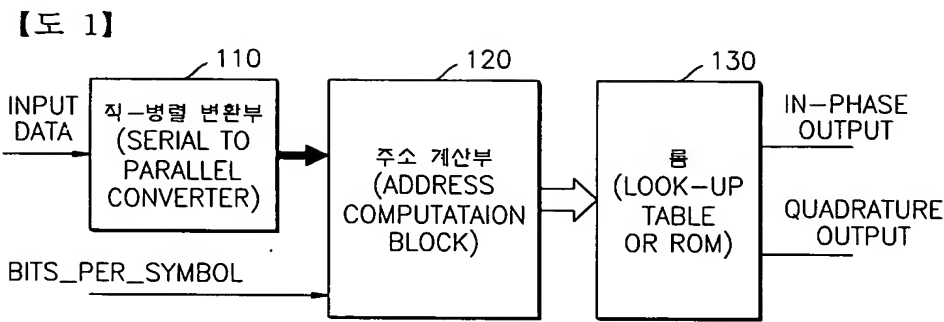
상기 이동시킨 값 중에서 하위 6 비트의 출력에서 5 내지 2번째 비트의 값과 이진값 '00'을 연결시켜 만든 데이터와,

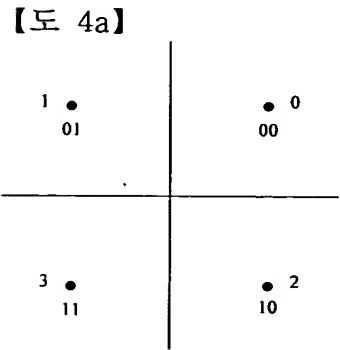
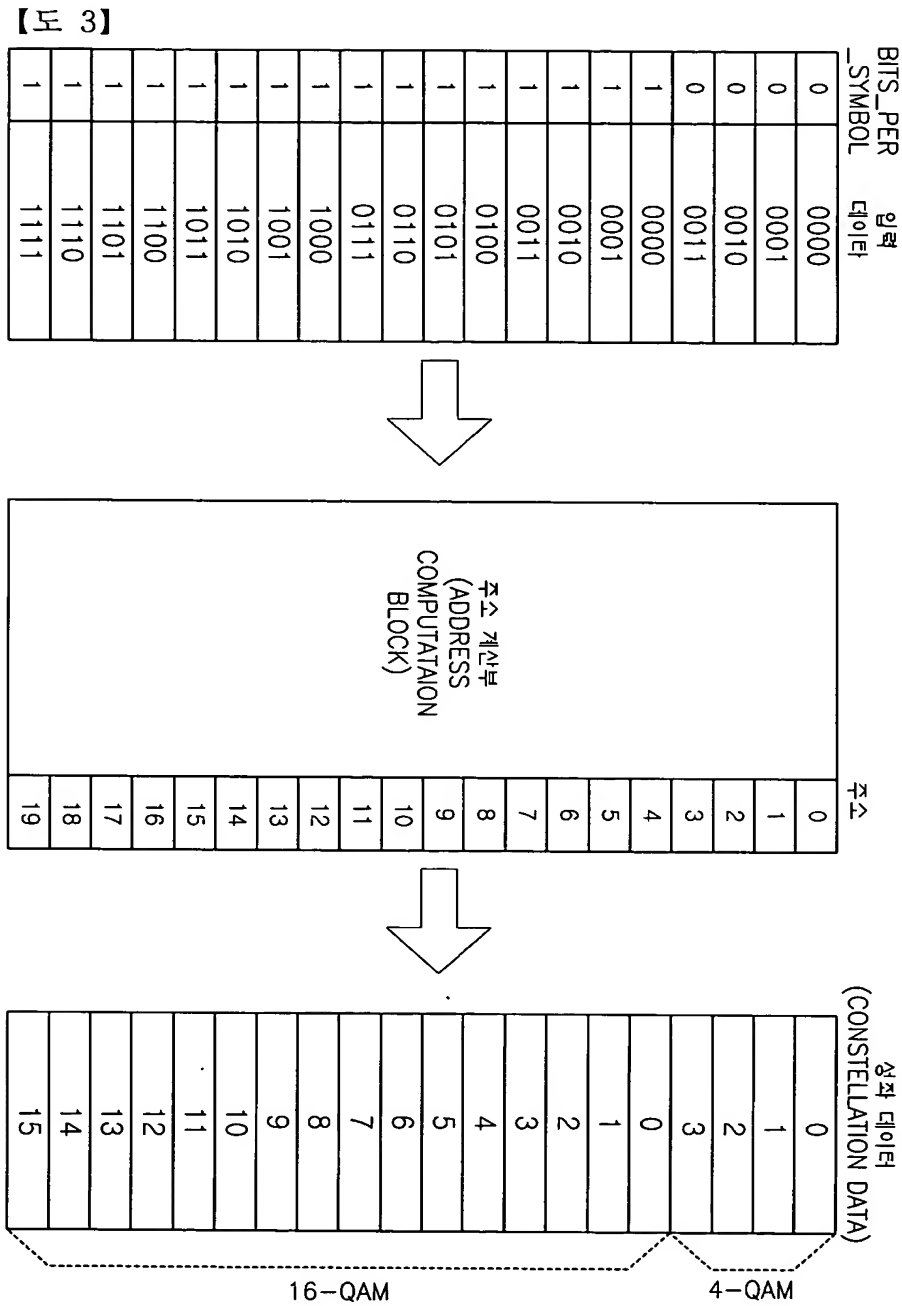
상기 이동시킨 값 중에서 하위 6 비트의 출력에서 5 내지 1번째 비트의 값과 이진값 '0'을 연결시켜 만든 데이터와, 이진값 '000000'을 입력받고, 상기 베이스 어드레스를 생성하는 단계에서 출력 신호를 선택단자로 입력받아 덧셈하는 단계를 포함하는 것을 특징으로 하는 성과 매핑 방법.

【청구항 15】

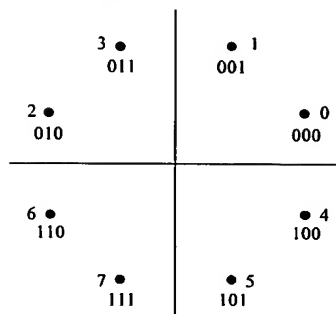
제10항에 기재된 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

【도면】

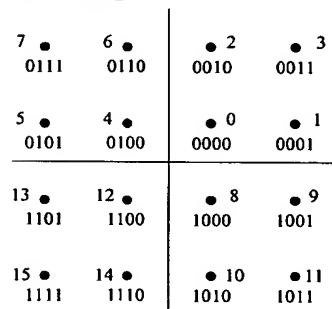




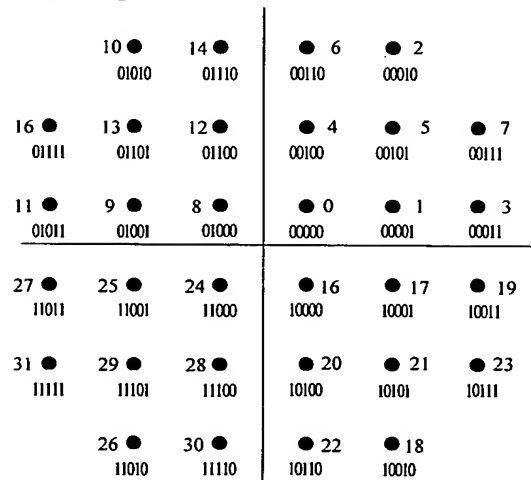
【도 4b】



【도 4c】



【도 4d】



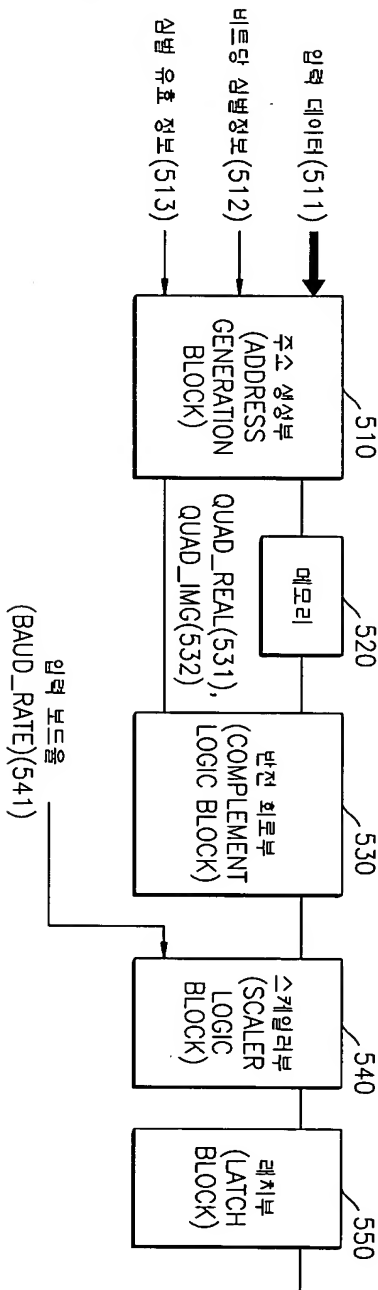
26 ● 011010	27 ● 011011	25 ● 011001	24 ● 011000	8 ● 001000	9 ● 001001	11 ● 001011	10 ● 001010
30 ● 011110	31 ● 011111	29 ● 011101	28 ● 011100	12 ● 001100	13 ● 001101	15 ● 001111	14 ● 001110
22 ● 010110	23 ● 010111	21 ● 010101	20 ● 010100	4 ● 000100	5 ● 000101	7 ● 000111	6 ● 000110
18 ● 010010	19 ● 010011	17 ● 010001	16 ● 010000	0 ● 000000	1 ● 000001	3 ● 000011	2 ● 000010
50 ● 110010	51 ● 110011	49 ● 110001	48 ● 110000	32 ● 100000	33 ● 100001	35 ● 100011	34 ● 100010
54 ● 110110	55 ● 110111	53 ● 110101	52 ● 110100	36 ● 100100	37 ● 100101	39 ● 100111	38 ● 100110
62 ● 111110	63 ● 111111	61 ● 111101	60 ● 111100	44 ● 101100	45 ● 101101	47 ● 101111	46 ● 101110
58 ● 111010	59 ● 111011	57 ● 111001	56 ● 111000	40 ● 101000	41 ● 101001	43 ● 101011	42 ● 101010

44 ● 0101100	45 ● 0101101	61 ● 0111101	60 ● 0111100	28 ● 0011100	29 ● 0011101	13 ● 0001101	12 ● 0001100
36 ● 0100100	37 ● 0100101	53 ● 0110101	52 ● 0110100	20 ● 0010100	21 ● 0010101	5 ● 0000101	4 ● 0000100
55 ● 0110111	54 ● 0110110	50 ● 0110010	51 ● 0110011	49 ● 0110001	48 ● 0110000	16 ● 0010000	17 ● 0010001
63 ● 0111111	62 ● 0111110	58 ● 0111010	59 ● 0111011	57 ● 0111001	56 ● 0111000	19 ● 0010011	18 ● 0010010
47 ● 0101111	46 ● 0101110	42 ● 0101010	43 ● 0101011	41 ● 0101001	40 ● 0101000	22 ● 0010110	23 ● 0010111
39 ● 0100111	38 ● 0100110	34 ● 0100010	35 ● 0100011	33 ● 0100001	32 ● 0100000	24 ● 0011000	25 ● 0011001
103 ● 1100111	102 ● 1100110	98 ● 1100010	99 ● 1100011	97 ● 1100001	96 ● 1100000	27 ● 0011011	26 ● 0011010
111 ● 1101111	110 ● 1101110	106 ● 1101010	107 ● 1101011	105 ● 1101001	104 ● 1101000	30 ● 0011110	31 ● 0011111
127 ● 1111111	126 ● 1111110	122 ● 1111010	123 ● 1111011	121 ● 1111001	120 ● 1111000	8 ● 0001000	9 ● 0001001
119 ● 1110111	118 ● 1110110	114 ● 1110010	115 ● 1110011	113 ● 1110001	112 ● 1110000	11 ● 0001011	10 ● 0001010
100 ● 1100100	101 ● 1100101	117 ● 1110101	116 ● 1110100	0 ● 0000000	1 ● 0000001	14 ● 0001110	15 ● 0001111
108 ● 1101100	109 ● 1101101	125 ● 1111101	124 ● 1111100	64 ● 1000000	65 ● 1000001	2 ● 0000010	3 ● 0000011
				72 ● 1001000	73 ● 1001001	6 ● 0000110	7 ● 0000111
				88 ● 1011000	89 ● 1011001	67 ● 1000011	66 ● 1000010
				80 ● 1010000	81 ● 1010001	70 ● 1000110	71 ● 1000111
				84 ● 1010100	85 ● 1010101	74 ● 1001011	75 ● 1001010
				92 ● 1011100	93 ● 1011101	78 ● 1001110	79 ● 1001111
						94 ● 1011110	95 ● 1011111
						86 ● 1010110	87 ● 1010111
						82 ● 1010010	83 ● 1010011
						69 ● 1000101	68 ● 1000100

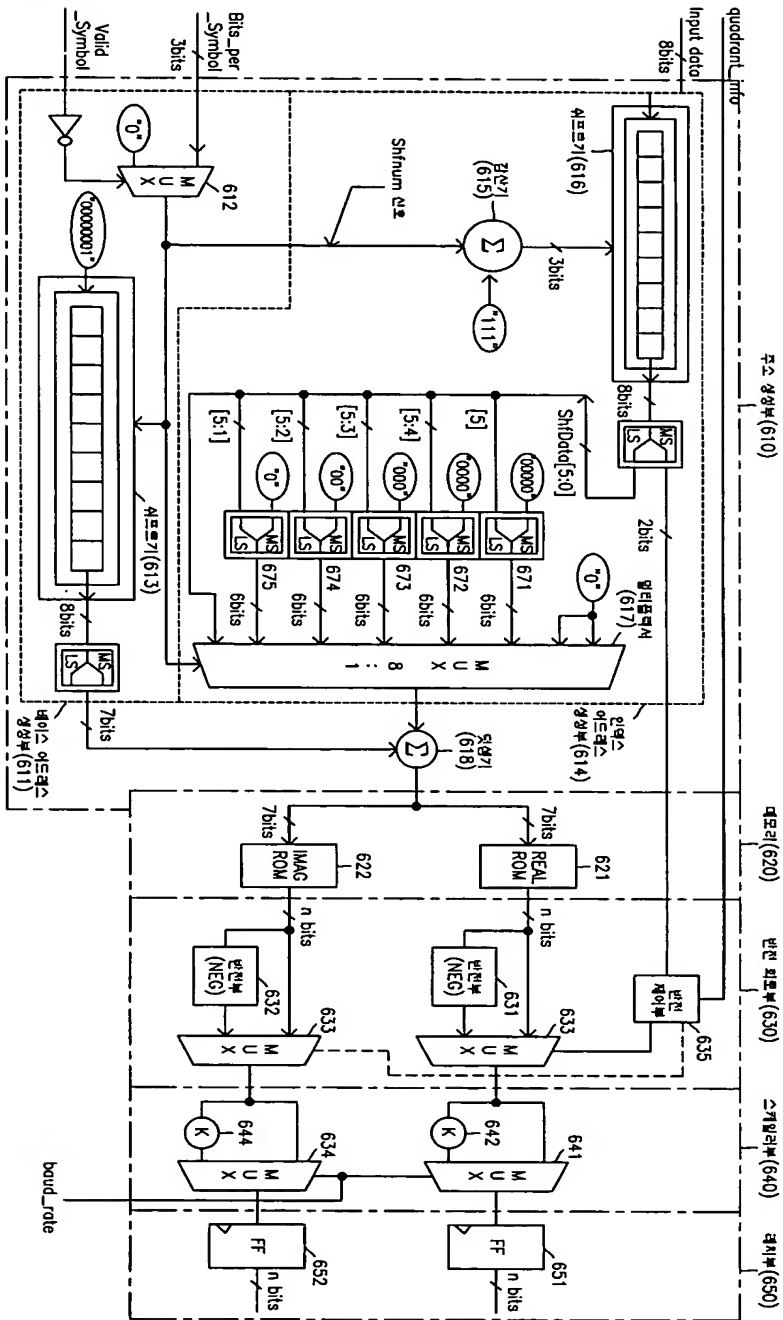
【도 4g】

100 ●	101 ●	103 ●	102 ●	98 ●	99 ●	97 ●	96 ●	32 ●	33 ●	35 ●	34 ●	38 ●	39 ●	37 ●	36 ●
01100000	01100001	01100011	01100010	01100000	01100001	01100000	01100000	00000000	00000001	00000011	00000010	00000010	00000011	00000001	00000000
108 ●	109 ●	111 ●	110 ●	106 ●	107 ●	105 ●	104 ●	40 ●	41 ●	43 ●	42 ●	46 ●	47 ●	45 ●	44 ●
01100100	01100101	01100111	01100110	01100100	01100101	01100100	01100100	00000000	00000001	00000011	00000010	00000010	00000011	00000001	00000000
124 ●	125 ●	127 ●	126 ●	122 ●	123 ●	121 ●	120 ●	56 ●	57 ●	59 ●	58 ●	62 ●	63 ●	61 ●	60 ●
01111000	01111001	01111011	01111010	01111000	01111001	01111000	01111000	00011000	00011001	00011011	00011010	00011010	00011011	00011001	00011000
116 ●	117 ●	119 ●	118 ●	114 ●	115 ●	113 ●	112 ●	48 ●	49 ●	51 ●	50 ●	54 ●	55 ●	53 ●	52 ●
01110000	01110001	01110011	01110010	01110000	01110001	01110000	01110000	00011000	00011001	00011011	00011010	00011010	00011011	00011001	00011000
84 ●	85 ●	87 ●	86 ●	82 ●	83 ●	81 ●	80 ●	16 ●	17 ●	19 ●	18 ●	22 ●	23 ●	21 ●	20 ●
01010000	01010001	01010011	01010010	01010000	01010001	01010000	01010000	00001000	00001001	00001011	00001010	00001010	00001011	00001001	00001000
82 ●	83 ●	95 ●	94 ●	90 ●	91 ●	89 ●	88 ●	24 ●	25 ●	27 ●	26 ●	30 ●	31 ●	29 ●	28 ●
01011000	01011001	01011011	01011010	01011000	01011001	01011000	01011000	00011000	00011001	00011011	00011010	00011010	00011011	00011001	00011000
76 ●	77 ●	79 ●	78 ●	74 ●	75 ●	73 ●	72 ●	8 ●	9 ●	11 ●	10 ●	14 ●	15 ●	13 ●	12 ●
01001000	01001001	01001011	01001010	01001000	01001001	01001000	01001000	00001000	00001001	00001011	00001010	00001010	00001011	00001001	00001000
68 ●	69 ●	71 ●	70 ●	66 ●	67 ●	65 ●	64 ●	0 ●	1 ●	3 ●	2 ●	6 ●	7 ●	5 ●	4 ●
01000000	01000001	01000011	01000010	01000000	01000001	01000000	01000000	00000000	00000001	00000011	00000010	00000010	00000011	00000001	00000000
196 ●	197 ●	199 ●	198 ●	194 ●	195 ●	193 ●	192 ●	128 ●	129 ●	131 ●	130 ●	134 ●	135 ●	133 ●	132 ●
11000000	11000001	11000011	11000010	11000000	11000001	11000000	11000000	10000000	10000001	10000011	10000010	10000010	10000011	10000001	10000000
204 ●	205 ●	207 ●	206 ●	202 ●	203 ●	201 ●	200 ●	136 ●	137 ●	139 ●	138 ●	142 ●	143 ●	141 ●	140 ●
11001000	11001001	11001011	11001010	11001000	11001001	11001000	11001000	10001000	10001001	10001011	10001010	10001010	10001011	10001001	10001000
220 ●	221 ●	223 ●	222 ●	218 ●	219 ●	217 ●	216 ●	152 ●	153 ●	155 ●	154 ●	158 ●	159 ●	157 ●	156 ●
11011000	11011001	11011011	11011010	11011000	11011001	11011000	11011000	10011000	10011001	10011011	10011010	10011010	10011011	10011001	10011000
212 ●	213 ●	215 ●	214 ●	210 ●	211 ●	209 ●	208 ●	144 ●	145 ●	147 ●	146 ●	150 ●	151 ●	149 ●	148 ●
11000000	11000001	11000011	11000010	11000000	11000001	11000000	11000000	10001000	10001001	10001011	10001010	10001010	10001011	10001001	10001000
244 ●	245 ●	247 ●	246 ●	242 ●	243 ●	241 ●	240 ●	176 ●	177 ●	179 ●	178 ●	182 ●	183 ●	181 ●	180 ●
11100000	11100001	11100011	11100010	11100000	11100001	11100000	11100000	10010000	10010001	10010011	10010010	10010010	10010011	10010001	10010000
252 ●	253 ●	255 ●	254 ●	250 ●	251 ●	249 ●	248 ●	184 ●	185 ●	187 ●	186 ●	190 ●	191 ●	189 ●	188 ●
11111000	11111001	11111011	11111010	11111000	11111001	11111000	11111000	10011000	10011001	10011011	10011010	10011010	10011011	10011001	10011000
236 ●	237 ●	239 ●	238 ●	234 ●	235 ●	233 ●	232 ●	168 ●	169 ●	171 ●	170 ●	174 ●	175 ●	173 ●	172 ●
11100000	11100001	11100011	11100010	11100000	11100001	11100000	11100000	10001000	10001001	10001011	10001010	10001010	10001011	10001001	10001000
228 ●	229 ●	231 ●	230 ●	226 ●	227 ●	225 ●	224 ●	160 ●	161 ●	163 ●	162 ●	166 ●	167 ●	165 ●	164 ●
11100000	11100001	11100011	11100010	11100000	11100001	11100000	11100000	10000000	10000001	10000011	10000010	10000010	10000011	10000001	10000000

【도 5】



【도 6】



【도 7】

Bits-per-Symbol	유효 입력 데이터 비트 7th → 0th bit (X: not valid)
"000"	"XXXXXXXX"
"001"	"XXXXXX00"
"010"	"XXXXX000"
"011"	"XXXX0000"
"100"	"XXX00000"
"101"	"XX000000"
"110"	"X0000000"
"111"	"00000000"

【도 8】

address	ROM Data
0	zero symbol
1	qpsk r(0)
2	8-psk r(0)
3	8-psk r(1)
4	16-qam r(0)
5	16-qam r(1)
6	16-qam r(2)
7	16-qam r(3)
⋮	
65	256-qam r(0)
66	256-qam r(1)
67	256-qam r(2)
⋮	
127	256-qam r(63)

address	ROM Data
0	zero symbol
1	qpsk I(0)
2	8-psk I(0)
3	8-psk I(1)
4	16-qam I(0)
5	16-qam I(1)
6	16-qam I(2)
7	16-qam I(3)
⋮	
65	256-qam I(0)
66	256-qam I(1)
67	256-qam I(2)
⋮	
127	256-qam I(63)

【도 9】

Bits-per-Symbol	유효 입력 데이터 비트 7th -> 0th bit (X: not valid)	Shfnum	7-Shfnum만큼 이동된 입력 데이터 (X: not valid)
"000"	"XXXXXXXX"	0	"00000000"
"001"	"XXXXXXQQ"	1	"QQ000000"
"010"	"XXXXXQQB"	2	"QQB00000"
"011"	"XXXXQQBB"	3	"QQBB0000"
"100"	"XXXQQBBB"	4	"QQBBB000"
"101"	"XXQQBBBB"	5	"QQBBBB00"
"110"	"XQQBBBBB"	6	"QQBBBBB0"
"111"	"QQBBBBBB"	7	"QQBBBBBB"

【도 10a】

S1=qi1, S0=qr0

분면	qi1	qr0	S1	S0
I	0	0	+	+
II	0	1	-	+
III	1	1	-	-
IV	1	0	+	-

【도 10b】

S1=qi1, S0=qr0'

분면	qi1	qr0	S1	S0
I	0	0	+	-
II	0	1	+	+
III	1	1	-	+
IV	1	0	-	-

【도 10c】

$$S1=qi1', S0=qr0'$$

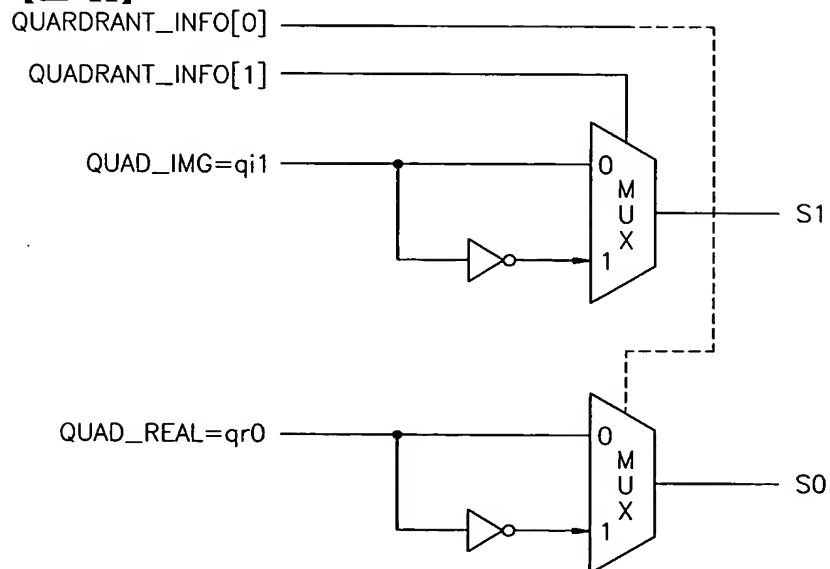
분면	qi1	qr0	S1	S0
I	0	0	-	-
II	0	1	-	+
III	1	1	+	+
IV	1	0	+	-

【도 10d】

$$S1=qi1', S0=qr0$$

분면	qi1	qr0	S1	S0
I	0	0	-	+
II	0	1	-	-
III	1	1	+	-
IV	1	0	+	+

【도 11】



【도 12】

